CLAIMS

1. A method of translating binary code instructions from a source format to a target format for processing by a target processor, said method comprising the steps of:

a) Identifying a source instruction;

b) Selecting a translation template corresponding to said identified source instruction, said template providing a set of target format instructions semantically equivalent to said identified source instruction; and

c) Translating said identified instruction in accordance with said template; and

d) Outputting said translated instruction for processing by said target processor.

2. A method according to claim 1 in which said source and target instructions include a control part and a data part and said control part being used in said identification step to identify an instruction.

3. A method according to claim 2 in further comprising a transformation step in which said data part from said source instruction is transformed into said corresponding data part or parts of said set of target format instructions.

4. A method according to claim 3 in which said transformation step is carried out in accordance with a bit filling routine associated with said template.

5. A method according to claim 4 in which said bit filling routine is uniquely associated with said template.

6. A method according to claim 3 in which said transformation step is arranged to transform data of one type of endianness to data of another type of endianness.

7. A method according to claim 2 in which said source instruction control parts are each concatenated to provide a unique identifier and said templates are indexed in accordance with said identifiers.

8. A method according to claim 7 in which said templates are indexed by said unique identifiers in a look up table.

9. A method according to claim 1 in which said translation is carried out at runtime of an emulated application program.

10. A method according to claim 1 in which said templates are provided by software procedure calls.

11. A method according to claim 1 in which said source format is 32 bit and said target format is 64 bit.

12. A method according to claim 1 in which said source format is PA-RISC code and said target format is Itanium™ code.

13. Apparatus for translating binary code instructions from a source format to a target format for processing by a target processor, the apparatus comprising:

a) An instruction identifier for identifying a source instruction;
b) A template selector for selecting a translation template corresponding to said identified source instruction, said translation template providing a set of target format instructions semantically equivalent to said identified source instruction; and
c) A translator for translating said identified instruction in accordance with said template; and
d) An output buffer for outputting said translated instruction for processing by said target processor.

14. Apparatus according to claim 13 in which said source and target instructions include a control part and a data part and said instruction identifier uses said control part to identify an instruction.

15. Apparatus according to claim 14 in which in said translator is operable to transform said data part from said source instruction into said corresponding data part or parts of said set of target format instructions.

16. Apparatus according to claim 15 in which said transformation is carried out in accordance with a bit filling routine associated with said template.

17. Apparatus according to claim 16 in which said bit filling routine is uniquely associated with said template.

18. Apparatus according to claim 15 in which translator is operable to transform data of one type of endianness into data of another type of endianness.

19. Apparatus according to claim 14 in which said source instruction control parts are concatenated to provide a unique identifier and said templates are indexed in accordance with said identifiers.

20. Apparatus according to claim 19 in which said templates are indexed by said unique identifiers in a look up table.

21. Apparatus according to claim 13 in which said translation is carried out at runtime of an emulated application program.

22. Apparatus according to claim 13 in which said templates are provided by software procedure calls.

23. Apparatus according to claim 13 in which said source code has a 32 bit format and said target code has a 64 bit format.

24. Apparatus according to claim 13 in which said source code is PA-RISC code and said target code is Itanium™ code.

25. A template for use in a binary code translator for translating binary code instructions from a source format to a target format for processing by a target processor, said template comprising:

a) A template identifier uniquely associating said template to a source instruction; and

b) A set of instructions in said target format semantically equivalent to said source instruction.

26. A template according to claim 25 in which said source and target instructions include a control part and a data part and said template identifier is derived from said control part of said source instruction.

27. A template according to claim 26 in which in said template is associated with a set of instructions to transform said data part from said source instruction into said corresponding data part or parts of said set of target format instructions.

28. A template according to claim 27 in which said transformation is carried out in accordance with a bit filling routine associated with said template.

29. A template according to claim 28 in which said bit filling routine is uniquely associated with said template.

30. A template according to claim 26 in which said template identifier is created by the concatenation of said control part of said source instruction.

31. A template according to claim 25 in which said source code has a 32 bit format and said target code has a 64 bit format.

32. A template according to claim 25 in which said source code is PA-RISC code and said target code is Itanium™ code.

33 A computer program for translating binary code instructions from a source format to a target format for processing by a target processor, in accordance with the method of claim 1

34. A computer program according to claim 33 in which said templates are implemented as routines in said computer program.

35. A computer program according to claim 33 operable to carry out said translation at said runtime of said source binary code.